



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,055	11/25/2003	Konstantin Melamed	SYMB-001UTL	4193
7590 05/23/2006				
Konstantin Melamed SymbioWare, LLC Suite 131 2977 Ygnacio Valley Road Walnut Creek, CA 94598			EXAMINER LU, KUEN S	
			ART UNIT 2167	PAPER NUMBER
DATE MAILED: 05/23/2006				

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/720,055	Applicant(s) MELAMED ET AL.	
	Examiner Kuen S. Lu	Art Unit 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 25 November 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-22 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-22 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 25 November 2003 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This Action is responsive to Applicant's Application, filed November 25, 2003.

Priority

2. Applicant's claim for the benefit of a prior-filed application U.S. Provisional Patent Application 60/430,378 filed December 3, 2002, under 35 U.S.C. 119(e) or under 35 U.S.C. 120, 121, or 365(c) is acknowledged.

Drawings

3. New corrected drawings in compliance with 37 CFR 1.121(d) are required in this application because Figs. 4-14 filed 11/25/2003 contain un-cleaned marks and unreadable labels in shaded areas, for example, Figs. 3-4 and 9-17. Applicant is advised to employ the services of a competent patent draftsman outside the Office, as the U.S. Patent and Trademark Office no longer prepares new drawings. The corrected drawings are required in reply to the Office action to avoid abandonment of the application. The requirement for corrected drawings will not be held in abeyance. **Please note** Applicant is required to submit acceptable corrected drawings within the time period set in the Office action. See 37 CFR 1.85(a). Failure to take corrective action within the set period will result in ABANDONMENT of the application.

Specification

4. Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means" and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need

for consulting the full patent text for details.

The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc.

5. The abstract of the disclosure is objected to because the Abstract contains the phrase "each test case and/or test step" which can not determine the definite content of the abstract. Appropriate correction is required. See MPEP § 608.01(b), CFR 1.72(a) and MPEP § 606.

6. The disclosure is objected to because of the following informalities: At Page 1, line 4 where USSN in "USSN 60/430,378" is unclear. Examiner interprets as "U.S. Provisional Patent Application Number". Appropriate correction is required.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless -

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 1-19 are rejected under 35 U.S.C. 102(b) as anticipated by Glenn et al.

(IPsec-WIT : The NISP IPsec Web-based Interoperability Test System, National Institute of Standards and Technology, 2000, hereafter "Glenn").

As per claim 1, Glenn teaches "web-interface mediated testing of a software application" (See Page 147 Abstract where IPsec-WIT is a WEB interoperability experimental test system without requiring distribution of test system software or relocation of test system) comprising:

"(a) selecting a test case for execution on a host machine using a web-interface provided by an application server, wherein said test case comprises one or more test steps" (See Fig. 1 and Page 151, right column, hereafter "r. col.", lines 23-27 where user tests new cases on IPsec-WIT tester integrated with certificate server and added configuration options, and test steps executed in according to the test sessions defined in the test script);

"(b) selecting the host machine on which to run said test case using said web-interface provided by said application server, said host machine comprising an automation tool and a whole or partial copy of the computer readable code of the software being tested" (See Figs. 1-2; Page 148, left column, hereafter "l. col.", lines 13-18 and Page 151, r. col., lines 23-27 where user tests new cases on a WEB interfaced and a more flexible than a fully automated system, the IPsec-WIT test tool further integrated with selected certificate server and added configuration options);

"(c) encoding a test case file comprising the one or more test steps of said selected test case and the name of an automation tool GUI environment file" (See Fig. 2; Page 148, r. col., lines 15-20 and Page 149, l. col., lines 1-7 and r. col., lines 26-35 where state files contain test control parameters are maintained for test configuration, and the test

environment is completely web-based providing user to control and interface for the test system, and to convey test configuration requirements);

“(d) transmitting said test case file from said application server to said selected host machine” (See Fig. 1; Page 148, r. col., lines 4-5; Page 149, r. col., lines 43-47; Page 150, l. col., lines 2-14 and Page 151, l. col., lines 33-40 where web based test system is configured with HTTP server, Test Engine, implementation under test and user communicate on internet and test cases are written in HTML for being to convey implementation requirements);

“(e) receiving said test case file and decoding the contents thereof; wherein said decoding comprises generating a test script by parsing test step syntax recognized by said automation tool from said test case file and receiving a copy of the automation tool GUI environment file” (See Fig. 2 and Page 149, r. col., lines 43-52 where Perl's capabilities of executing system commands and parsing are utilized to develop test language for being converted into system commands and further for self-documented and executed); and

“(f) loading said test script, said automation tool GUI environment file, and said whole or partial copy of the computer readable code of the software being tested into said automation tool and executing said loaded test script using said automation tool thereby testing a software application” (See Fig. 2; Page 148, r. col., lines 15-20; Page 149, l. col., lines 1-7 and r. col., lines 26-35 and Page 150, l. col., lines 2-14 where state files contain test control parameters are maintained for test configuration, and the test environment is completely web-based providing user to control and interface for the test

system, and to convey test configuration requirements. Further the test description language made up of system directives and Unix commands are sequentially processed and test results being conveyed to user).

As per claim 12, Glenn teaches "web-interactive software testing" (See Page 147 Abstract where IPsec-WIT is a WEB interoperability experimental test system without requiring distribution of test system software or relocation of test system) comprising: "(a) an application server suitable as a web server operatively coupled to a database wherein said application server comprises a computer readable storage medium having computer readable code means for providing a web interface to one or more users, computer readable code means for populating said web-interface with data gathered from said database, and computer readable code means for populating said web-interface with data received from one or more host machines" (See Figs. 1-2 Page 148, l. col., lines 13-18, r. col., lines 4-5; Page 149, r. col., lines 43-47; Page 150, l. col., lines 2-14 and Page 151, l. col., lines 33-40 and r. col., lines 23-27 wherein the IPsec-WIT test application server, storage is provided for test suites and state files, database of state files and HTTP server are coupled to test engine, user is WEB-interfaced to test new cases and data is received from IUT and HTTP server); "(b) one or more user machines in communication with said application server suitable, said user machines comprising computer readable storage media including computer readable code means for interacting with the web interface provided by said application server" (See Figs. 1-2 Page 148, l. col., lines 13-18, r. col., lines 4-5; Page 149, r. col.,

lines 43-47; Page 150, l. col., lines 2-14 and Page 151, l. col., lines 33-40 and r. col., lines 23-27 where user browses and navigate on and wherein the IPsec-WIT test application server, storage is provided for test suites and state files, database of state files and HTTP server are coupled to test engine, user is WEB-interfaced to test new cases and data is received from IUT and HTTP server); and

“(c) one or more host machines in communication with said application server, said one or more host machines comprising computer readable storage media including an automation tool, a full or partial copy of the computer readable program code of a software application to be tested, and computer readable code means for loading a test script, an automation tool GUI environment file, and the computer readable program code of said software application into said automation tool and commanding the execution of said test script by said automation tool” (See Figs. 1-2 and Page 149, r. col., lines 43-52; Page 148, r. col., lines 15-20; Page 149, l. col., lines 1-7, r. col., lines 26-35 and Page 150, l. col., lines 2-14 where Perl’s capabilities of executing system commands and parsing are utilized to develop test language for being converted into system commands and further for self-documented and executed, and where state files contain test control parameters are maintained for test configuration, and the test environment is completely web-based providing user to control and interface for the test system, and to convey test configuration requirements. Further the test description language made up of system directives and Unix commands are sequentially processed and test results being conveyed to user).

As per claim 2, Glenn further teaches “transmitting status information about said test case from said host machine to said application server and providing said status information to a user by way of said web interface” (See Page 150, r. col., lines 5-13 where test output is displayed back to the user in a single result page).

As per claim 3, Glenn further teaches “transmitting status information about one or more test steps of said test case from said host machine to said application server and providing said status information to a user by way of said web interface” (See Page 150, r. col., lines 5-13 where test output is displayed back to the user in a single result page, and the page includes packet dump of ping packets between systems).

As per claims 4 and 14, Glenn further teaches “test case file is selected from a flat file, markup language encoded file, XML file, HTML file, ASCII file, or XHTML file” (See Page 150, l. col., lines 1-4 where description language implemented for a test system is one of HTML extension).

As per claims 5 and 15, Glenn further teaches “scheduling the execution of said selected test case on said selected host machine, wherein said scheduling comprises a time and a date for executing said selected test case on said selected host machine” (See Page 147, r. col., lines 3-7 and 30-37 where time and order of tests are controlled by user and at Page 148, r. col., lines 4-20 where user has the control of diagnostic interface to the test system).

As per claim 6, Glenn further teaches “a querying of the application server by the host machine whether a test case has been assigned to the host machine” (See Page 147, r. col., lines 3-7 and 23-37 where test system is remotely accessed to support web-based and interactive testing in which time and order of tests are controlled by user).

As per claim 7, Glenn further teaches “host machine requesting from the application server the encoding and transmitting of a test case file based on the test case assigned to said host machine” (See Fig. 2; Page 148, r. col., lines 15-20 and Page 149, l. col., lines 1-7 and r. col., lines 26-35 where state files contain test control parameters are maintained for test configuration, and the test environment is completely web-based providing user to control and interface for the test system, and to convey test configuration requirements, and at Fig. 1; Page 148, r. col., lines 4-5; Page 149, r. col., lines 43-47; Page 150, l. col., lines 2-14 and Page 151, l. col., lines 33-40 where web based test system is configured with HTTP server, Test Engine, implementation under test and user communicate on internet and test cases are written in HTML for being to convey implementation requirements).

As per claims 8 and 16, Glenn further teaches “authoring a test case using said web interface wherein said authoring comprises adding steps or modifying existing steps by selecting from a plurality of GUI objects” (See Page 147, r. col., lines 20-22 where testing code change and re-testing are allowed).

As per claims 9 and 17, Glenn further teaches "modifying an action, procedure, or expected result corresponding to a selected GUI object" (See Page 147, r. col., lines 12-16 where formal testing procedures are established and at Page 148, r. col., lines 4-20 where user is provided with control and diagnostic interface to the test system through GUI interface tool).

As per claims 10 and 18, Glenn further teaches "GUI objects are preset GUI objects" (See Page 147, r. col., lines 12-16 where formal testing procedures are established and at Page 148, r. col., lines 4-20 where user is provided with control and diagnostic interface to the test system through GUI interface tool).

As per claim 11, Glenn further teaches "GUI objects are manually created and an action, procedure, or expected result corresponding to said manually created GUI object is entered using automation tool recognized syntax" (See Page 151, l. col., lines 33-40 where test cases are created tailor for a particular user and at Page 148, r. col., lines 4-20 where user is provided with control and diagnostic interface to the test system through GUI interface tool).

As per claim 13, Glenn further teaches the following:

“(a) computer readable program code means for enabling a user to select of a test case” (See Page 148, r. col., lines 15-20 where user is provided with control and diagnostic interface to the test system);

“(b) computer readable program code means for enabling a user to select a host machine on which to execute said selected test case” (See Page 148, r. col., lines 15-20 and Page 147, r. col., lines 27-30 where user is provided with control and diagnostic interface to the test system and the web-based system allows users to test products between bakeoffs);

“(c) computer readable program code means for encoding a test case file wherein said test case file comprises one or more of test steps of said selected test case and the name of an automation tool GUI environment file” (See Figs. 1-2 Fig. 1 and Page 151, right column, hereafter “r. col.”, lines 23-27 where user tests new cases on IPsec-WIT tester integrated with certificate server and added configuration options, and test steps executed in according to the test sessions defined in the test script); and

“(d) computer readable program code means for transmitting said encoded test case file to a host machine” (See Fig. 1; Page 148, r. col., lines 4-5; Page 149, r. col., lines 43-47; Page 150, l. col., lines 2-14 and Page 151, l. col., lines 33-40 where web based test system is configured with HTTP server, Test Engine, implementation under test and user communicate on internet and test cases are written in HTML for being to convey implementation requirements).

As per claim 19, Glenn further teaches the following:

“(a) computer readable program code means for querying said application server whether a test case has been scheduled for execution by said host machine” (See Page 147, r. col., lines 3-7 and 23-37 where test system is remotely accessed to support web-based and interactive testing in which time and order of tests are controlled by user);

“(b) computer readable program code means for requesting the encoding of and transmission of a test case file by the application server” (See Fig. 2; Page 148, r. col., lines 15-20 and Page 149, l. col., lines 1-7 and r. col., lines 26-35 where state files contain test control parameters are maintained for test configuration, and the test environment is completely web-based providing user to control and interface for the test system, and to convey test configuration requirements, and at Fig. 1; Page 148, r. col., lines 4-5; Page 149, r. col., lines 43-47; Page 150, l. col., lines 2-14 and Page 151, l. col., lines 33-40 where web based test system is configured with HTTP server, Test Engine, implementation under test and user communicate on internet and test cases are written in HTML for being to convey implementation requirements);

“(c) computer readable program code means for receiving said test case file” (See Fig. 2 and Page 149, r. col., lines 43-52 where Perl's capabilities of executing system commands and parsing are utilized to develop test language for being converted into system commands and further for self-documented and executed);

“(d) computer readable program code means for decoding said test case file, said decoding comprising generating a test script by parsing automation tool recognized syntax from said test case file, and resolving the name of an automation tool GUI environment file from said test case file” (See Fig. 2 and Page 149, r. col., lines 43-52

where Perl's capabilities of executing system commands and parsing are utilized to develop test language for being converted into system commands and further for self-documented and executed);

"(e) computer readable program code means for requesting the transmission of and receiving said automation tool GUI environment file from the application server" (See Fig. 2 and Page 149, r. col., lines 43-52 where Perl's capabilities of executing system commands and parsing are utilized to develop test language for being converted into system commands and further for self-documented and executed);

"(f) computer readable program code means for updating the application server with the status of a test case assigned to be executed on said host machine or the status of the one or more test steps included in said test case" (See Page 150, r. col., lines 5-13 where test output is displayed back to the user in a single result page, and the page includes packet dump of ping packets between systems).

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

10. Claims 20-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Glenn et al. (IPsec-WIT: The NISP IPsec Web-based Interoperability Test System, National Institute of Standards and Technology, 2000, hereafter "Glenn") in view of Hogan et al. (Information Technology Measurement and Testing Activities at NIST, Journal of Research of the National Institute of Standards and Technology, vol. 106, No. 1, January-February 2000, hereafter "Hogan").

As per claim 20, Glenn teaches "managing software testing requirements" (See Fig. 2; Page 148, r. col., lines 15-20 and Page 149, l. col., lines 1-7 and r. col., lines 26-35 where state files contain test control parameters are maintained for test configuration, and the test environment is completely web-based providing user to control and interface for the test system, and to convey test configuration requirements) comprising: "(a) providing a web interface wherein one or more users can create one or more requirements folders using said web interface (See Fig. 2; Page 148, r. col., lines 15-20 and Page 149, l. col., lines 1-7 and r. col., lines 26-35 where state files contain test control parameters are maintained for test configuration, and the test environment is completely web-based providing user to control and interface for the test system, and to convey test configuration requirements) and saving said one or more requirements folders" (See Fig. 2; Page 148, r. col., lines 15-20 and Page 149, l. col., lines 1-7 and r. col., lines 26-35 where state files contain test control parameters are maintained for test configuration and user utilizes a common web browser to navigate through directories of pre-defined test cases, configure test system, execute tests and view test results).

Glenn does not explicitly teach saving the requirements folders “in a relational database coupled to an application server that provides said web interface”, although Glenn teaches state files and directories as databases for the test to be based.

However, Hogan teaches using relational database to store images and images files to store images for testing algorithm encoding and decoding at Page 360, r. col. Lines 7-28.

It would have been obvious to one having ordinary skill in the art at the time of the Applicant's invention was made to combine Hogan's teaching on utilizing relational database for storing data or file with Glenn references because both references are directed to and part of measurement and testing activities conducted or proposed at NIST where the combined teaching would have enabled Glenn's interoperability testing to fully utilize the benefit of relational database technology to store and retrieve data in according to the test cases.

The combined teaching of the Glenn and Hogan references further teaches the following:

“(b) providing a web interface wherein one or more users can create one or more test cases using a web interactive authoring tool and saving said one or more test cases in a relational database coupled to an application server that provides said web interface” (See Glenn: Figs. 1-2, Page 148, l. col., lines 13-21 and Page 149, r. col., lines 36-52 where users in a community are able to use a WEB and internet based test system to conduct test cases dynamically instantiated for specific user, and Hogan: Page 360, r.

col. Lines 7-28 where relational database is utilized to store images and images files to store images for testing algorithm encoding and decoding); and

“(c) providing a web interface wherein one or more users can sort said one or more test cases within said one or more requirements folders” (See Glenn: Figs. 1-2, Page 148, l. col., lines 13-21 and Page 149, r. col., lines 36-52 and l. col., lines 1-7 where users in a community are able to use a WEB and internet based test system to conduct test cases dynamically instantiated for specific user and a common web browser to navigate through directories of pre-defined test cases, configure test system, execute tests and view test results).

As per claim 21, Glenn further teaches “a web interface wherein one or more users can create one or more sub-requirements folders using said web interface and saving said one or more sub-requirements folders in said one or more requirements folders; wherein both said requirements and said sub-requirements folders are saved in a relational database coupled to said application server that provides said web interface” (See Glenn: Figs. 1-2; Page 148, l. col., lines 13-18 and Page 151, r. col., lines 23-27 where user community tests new cases on a WEB interfaced and a more flexible than a fully automated system, the IPsec-WIT test tool further integrated with selected certificate server and added configuration options, and at Page 148, r. col., lines 15-20 and Page 149, l. col., lines 1-7 and r. col., lines 26-35 where state files contain test control parameters are maintained for test configuration and user utilizes a common

web browser to navigate through directories of pre-defined test cases, configure test system, execute tests and view test results).

11. Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over Glenn et al. (IPsec-WIT: The NISP IPsec Web-based Interoperability Test System, National Institute of Standards and Technology, 2000, hereafter "Glenn"), as applied to claim 1 above, and in view of Hogan et al. (Information Technology Measurement and Testing Activities at NIST, Journal of Research of the National Institute of Standards and Technology, vol. 106, No. 1, January-February 2000, hereafter "Hogan").

As per claim 22, Glenn teaches web-interface testing system as previously described in claim 1 rejection and further teaches "(a) a web interface wherein one or more users can create one or more requirements folders using said web interface (See Fig. 2; Page 148, r. col., lines 15-20 and Page 149, l. col., lines 1-7 and r. col., lines 26-35 where state files contain test control parameters are maintained for test configuration, and the test environment is completely web-based providing user to control and interface for the test system, and to convey test configuration requirements) and saving said one or more requirements folders" (See Fig. 2; Page 148, r. col., lines 15-20 and Page 149, l. col., lines 1-7 and r. col., lines 26-35 where state files contain test control parameters are maintained for test configuration and user utilizes a common web browser to navigate through directories of pre-defined test cases, configure test system, execute tests and view test results).

Glenn does not explicitly teach saving the requirements folders are saved "in a relational database coupled to an application server that provides said web interface", although Glenn teaches state files and directories as databases for the test to be based.

However, Hogan teaches using relational database to store images and images files to store images for testing algorithm encoding and decoding at Page 360, r. col. Lines 7-28.

It would have been obvious to one having ordinary skill in the art at the time of the Applicant's invention was made to combine Hogan's teaching on utilizing relational database for storing data or file with Glenn references because both references are directed to and part of measurement and testing activities conducted or proposed at NIST where the combined teaching would have enabled Glenn's interoperability testing to fully utilize the benefit of relational database technology to store and retrieve data in according to the test cases.

The combined teaching of the Glenn and Hogan references further teaches the following:

"(b) computer readable program code means for providing a web-interactive interface wherein one or more users can create one or more test cases using a web-interactive authoring tool and saving said one or more test cases in a relational database coupled to said application server that provides said web interface" (See Glenn: Figs. 1-2, Page 148, l. col., lines 13-21 and Page 149, r. col., lines 36-52 where users in a community are able to use a WEB and internet based test system to conduct test cases dynamically instantiated for specific user, and Hogan: Page 360, r. col. Lines 7-28

where relational database is utilized to store images and images files to store images for testing algorithm encoding and decoding); and

“(c) computer readable program code means for providing a web interface wherein one or more users can sort said one or more test cases within said one or more requirements folders” (See Glenn: Figs. 1-2, Page 148, l. col., lines 13-21 and Page 149, r. col., lines 36-52 and l. col., lines 1-7 where users in a community are able to use a WEB and internet based test system to conduct test cases dynamically instantiated for specific user and a common web browser to navigate through directories of pre-defined test cases, configure test system, execute tests and view test results).

Conclusions

12. The prior art made of record

U. Glenn et al.: IPsec-WIT : The NISP IPsec Web-based Interoperability Test System, National Institute of Standards and Technology, 2000

V. Hogan et al.: Information Technology Measurement and Testing Activities at NIST, Journal of Research of the National Institute of Standards and Technology, vol. 106, No. 1, January-February 2000

The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure.

A. U.S. Patent Application 2002/0095436

B. U.S. Patent Application 2002/0028430

Contact information

13. Any inquiry concerning this communication or earlier communications from the

Art Unit: 2167

examiner should be directed to Kuen S Lu whose telephone number is (571) 272-4114. The examiner can normally be reached on Monday-Friday (8:00 am-5:00 pm). If attempts to reach the examiner by telephone pre unsuccessful, the examiner's Supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

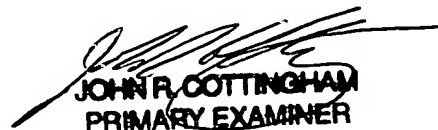
Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for Page 13 published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 886-217-9197 (toll-free).

Kuen S. Lu



Patent Examiner

May 18, 2006



JOHN R. COTTINGHAM
PRIMARY EXAMINER

